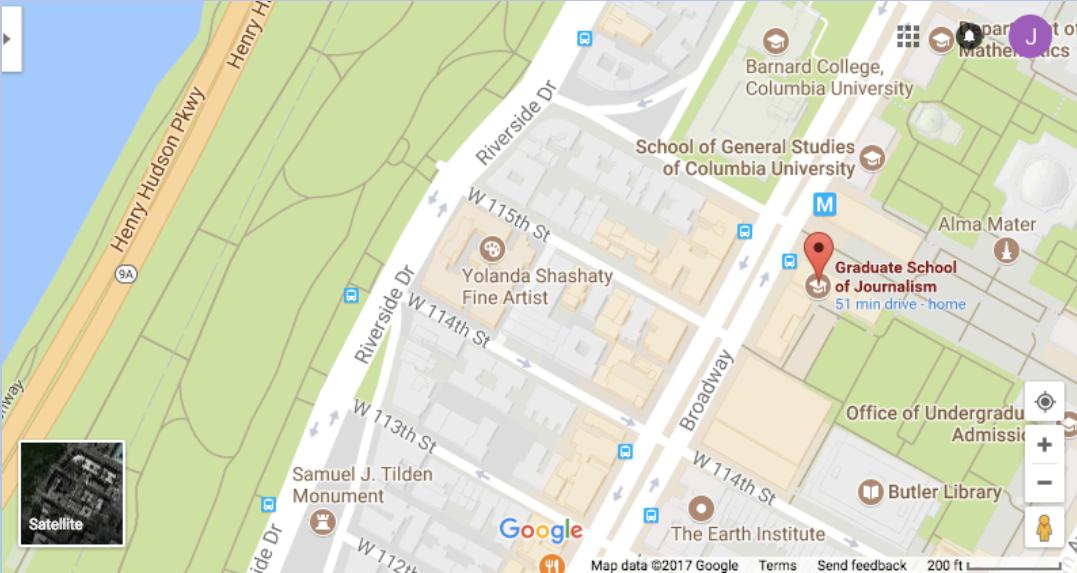


FRIDAY, JUNE 30

An introduction to Leaflet, a glance at other mapping tools, a lil' about
JavaScript, types of geographic data, methods of interaction

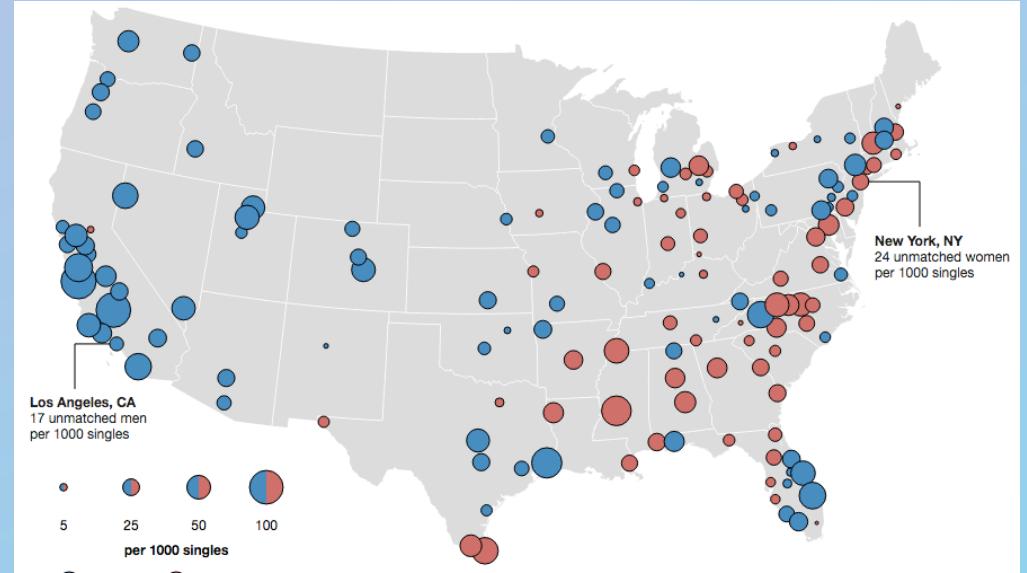
WHAT IS LEAFLET?

SLIPPY MAP



Interactive
Zooming, panning, dragging
Can go anywhere, usually
User customizes experience

NON-SLIPPY MAP



Usually can't zoom or pan
Can still be interactive!
But yes, can be a static map

+

-

WHY A SLIPPY MAP?



NYC's PLUTO data, Visualized



Data via [PLUTO](#) and [OpenStreetMap](#).
[MapBox](#). Inspired by Justin Palmer's [BKLYNR's Brooklyn map](#).

Many dates are estimates - buildings
the early 1900s often have a date of
[Brooklyn11211](#) for more details.

©

+

-

ABOUT THE USER

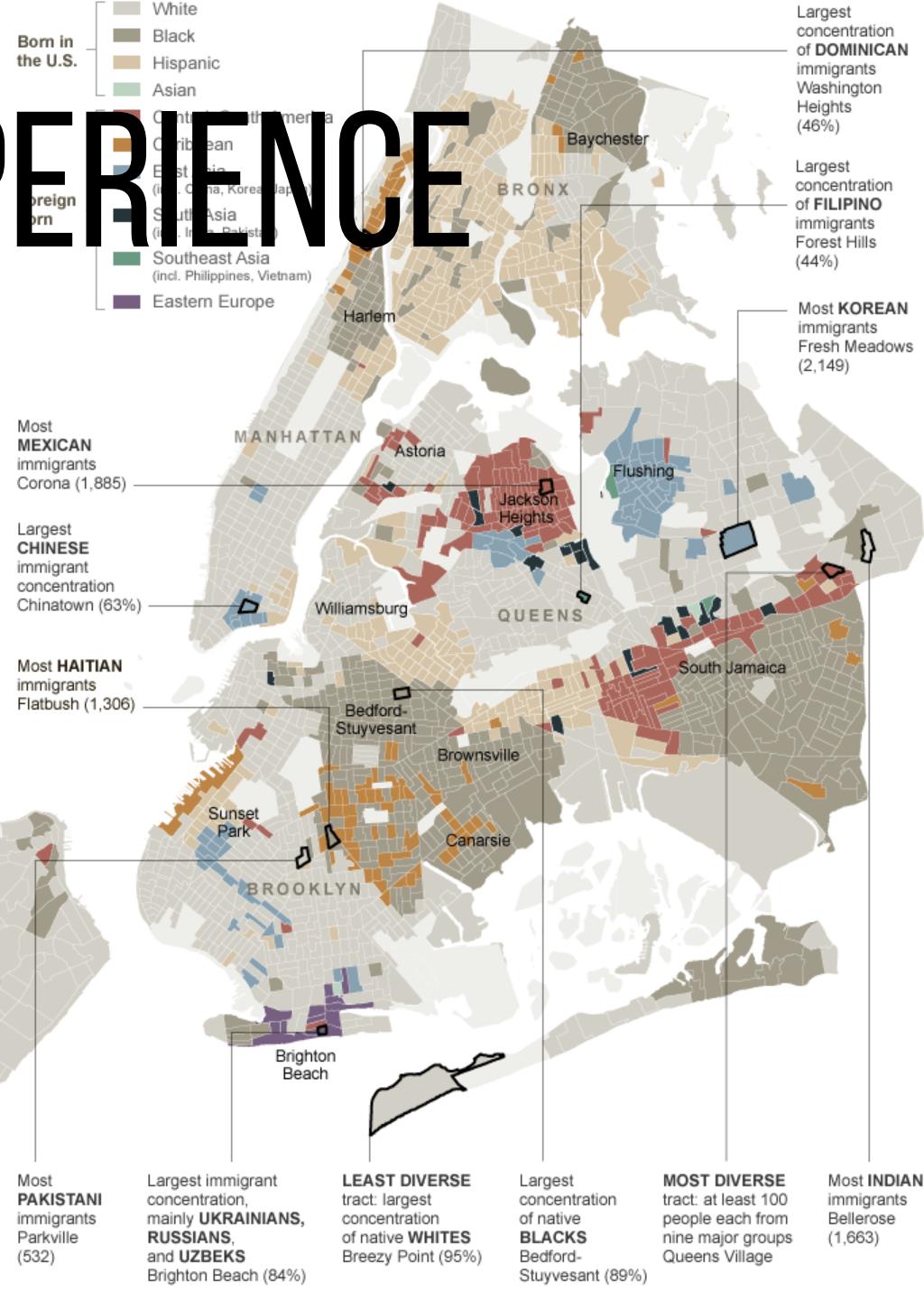
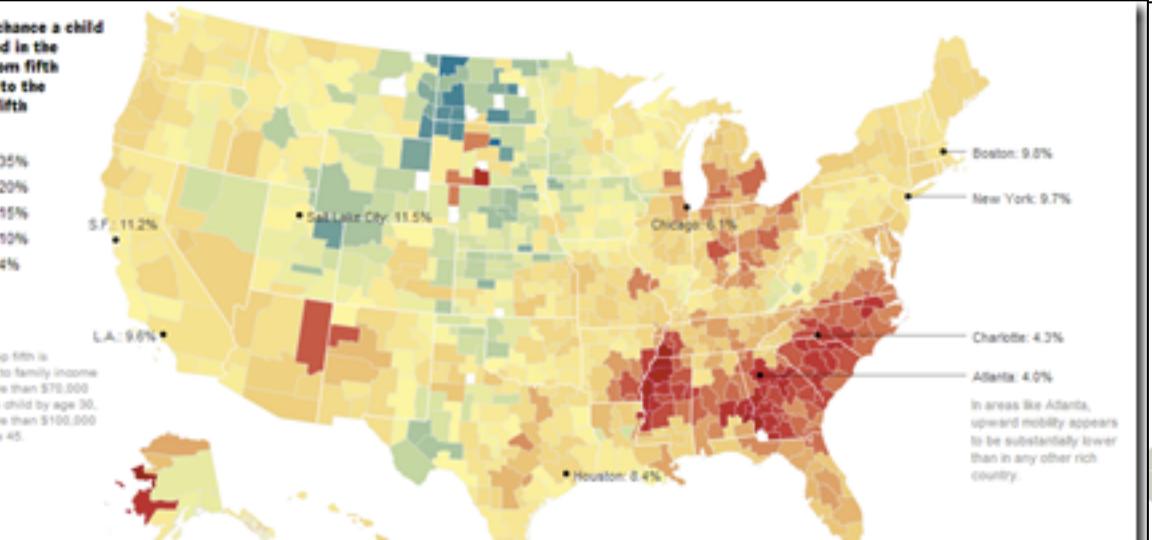
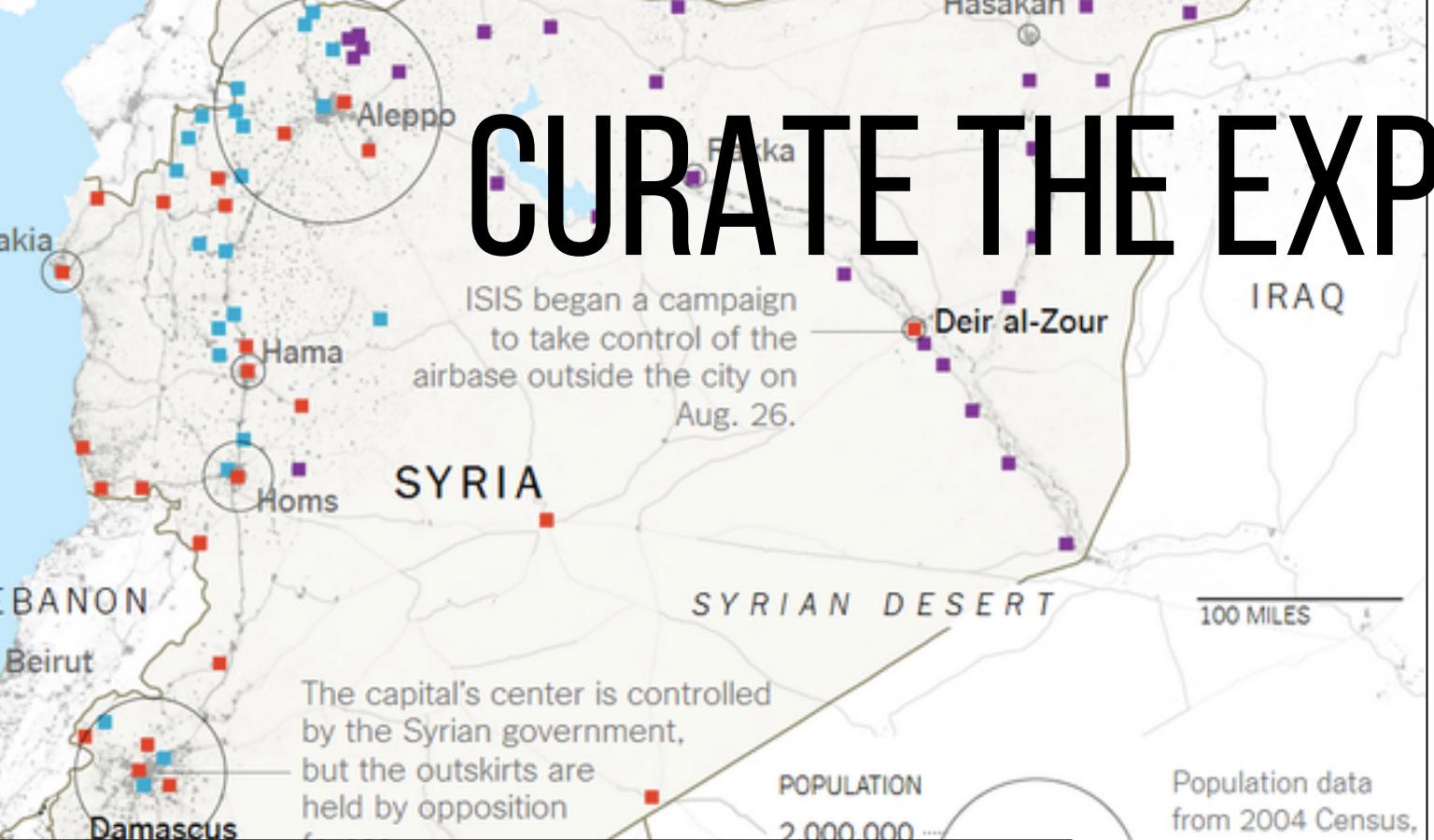
NYC's PLUTO data, Visualized

1830 1905

Data via [PLUTO](#) and [OpenStreetMap](#). Inspired by Justin Palmer's [BKLYNR's Brooklyn map](#).

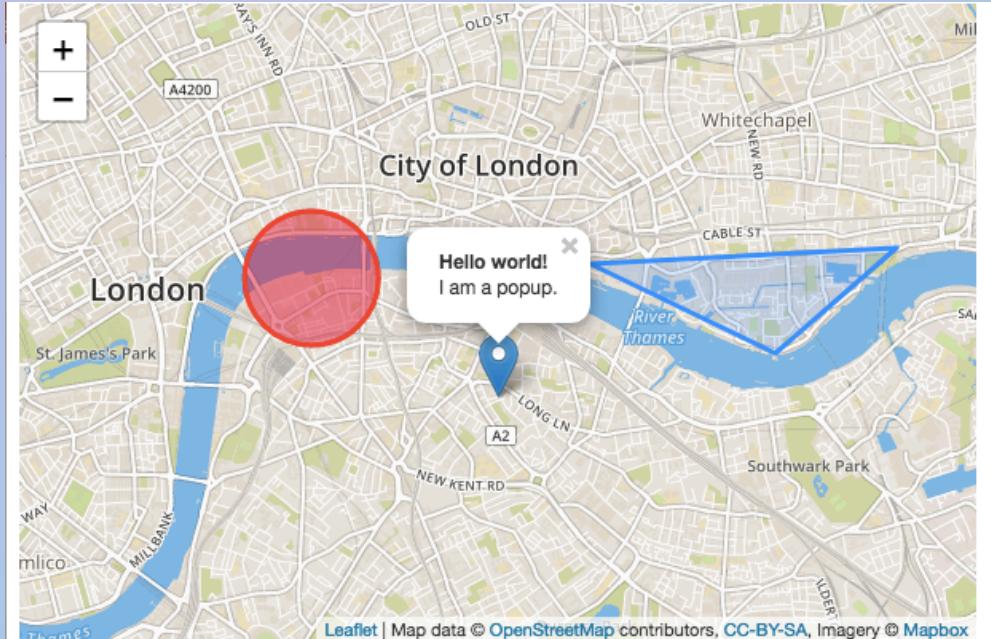
Many dates are estimates - buildings from the early 1900s often have a date of 1905. See [Brooklyn11211](#) for more details.

...EXCEPT THAT'S USUALLY A
BIG FAT LIE TOLD BY LAZY OR
UNSKILLED STORYTELLERS



JUST THINK ABOUT IT.

CUSTOM

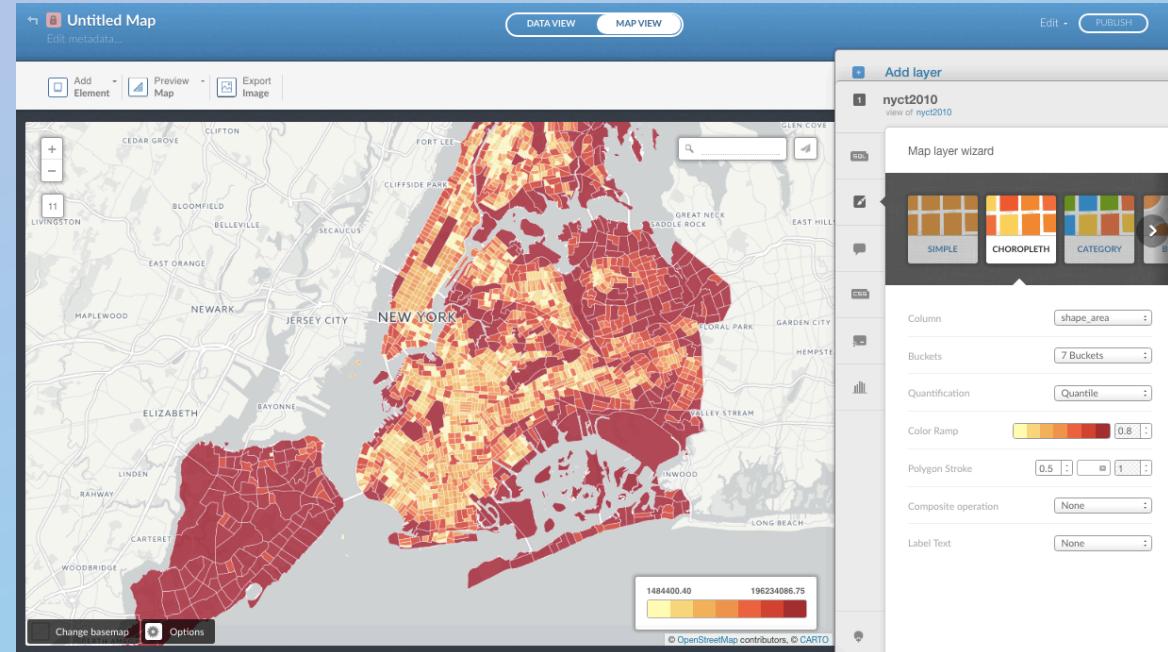


Leaflet

OpenLayers

*And also kind of MapBox.js,
MapBox GL JS, CARTO.js*

HOSTED/WIZARDS



Google Maps

CARTO

MapBox
ESRI

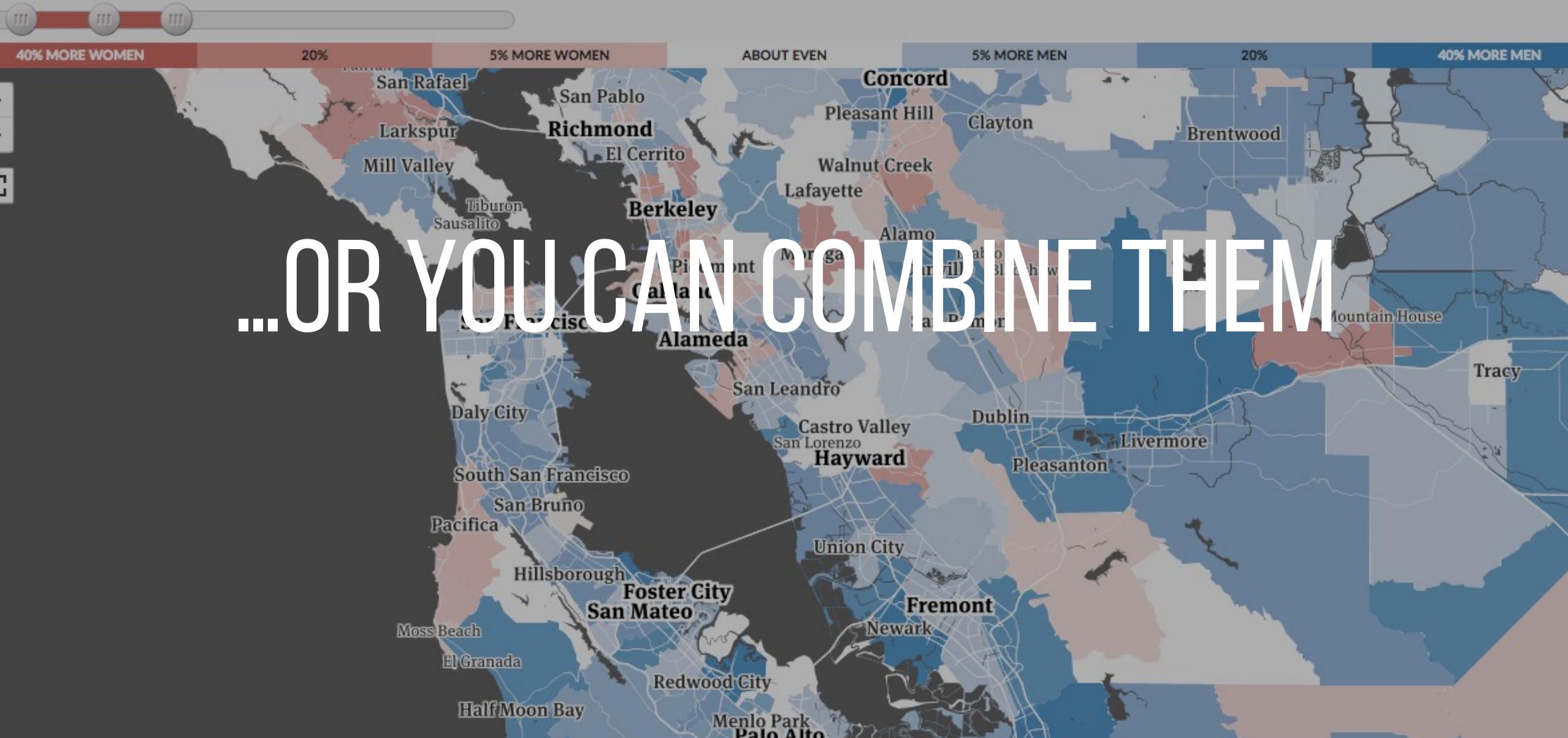
Bay Area Singles, 20-34

INSTRUCTIONS

Step 1: Slide the slider to the left around a bit.

Step 2: Get depressed, eat several cartons of ice cream while watching Netflix.

Check out our [favorite discoveries](#), get on our [super-cool newsletter](#), or [view this fullscreen?](#)



INTRO TO JAVASCRIPT

PYTHON

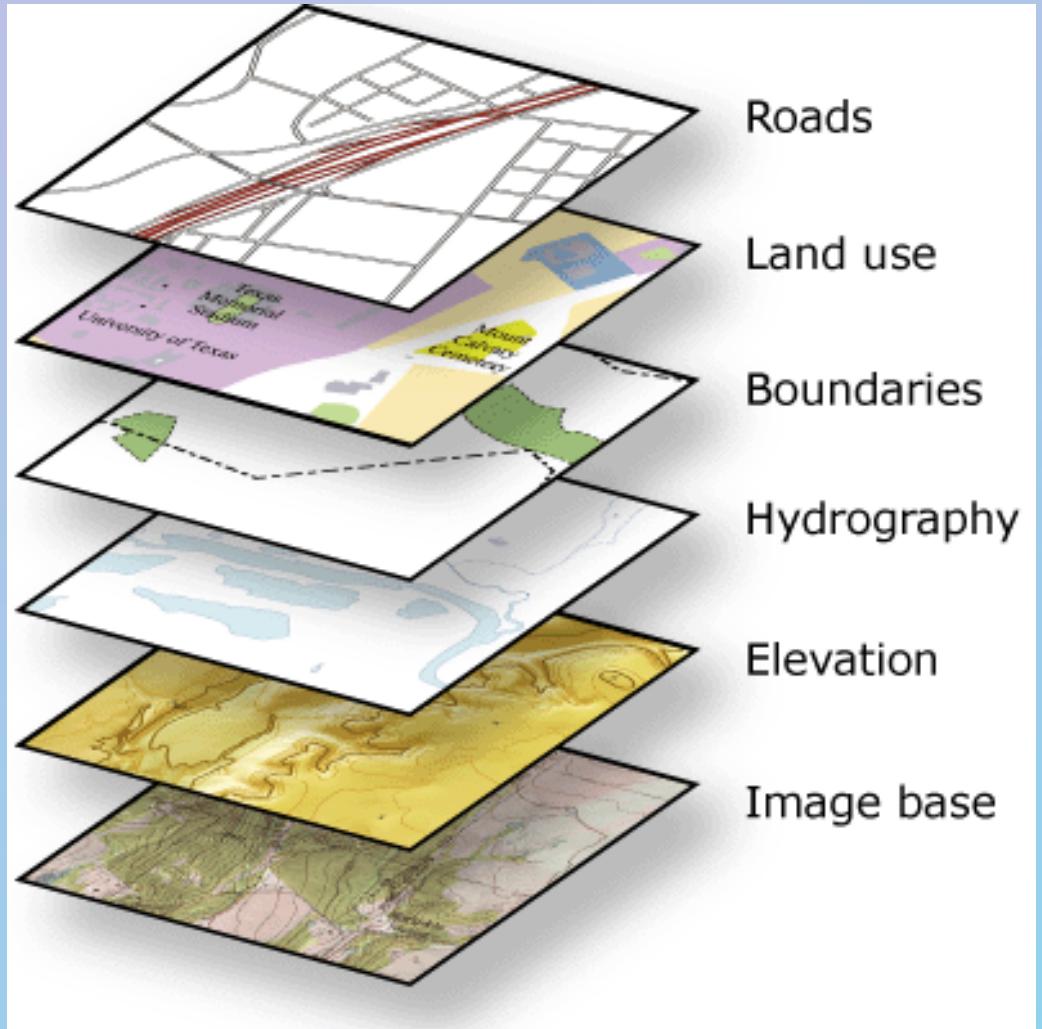
```
def hello(name):  
    print("Hello", name)  
  
# Lists  
nums = [3, 4, 5]  
print(len(nums))  
for number in nums:  
    print(number)  
  
# Dictionaries  
my_friend = {'name': 'B', 'age': 98}  
# Sending options to functions  
do_dance(thing, inplace=True)
```

JAVASCRIPT

```
function hello(name) {  
    console.log(name);  
}  
  
// Arrays  
var nums = [3, 4, 5];  
console.log(nums.length);  
for(i = 0; i < nums.length; i++) {  
    console.log(nums[i]);  
}  
  
// Objects or hashes  
var myFriend = {name: 'B', age: 98}  
// Sending optins to functions  
doDance(thing, {inplace: true})
```

WHAT GOES ON A MAP?

LAYERS UPON LAYERS

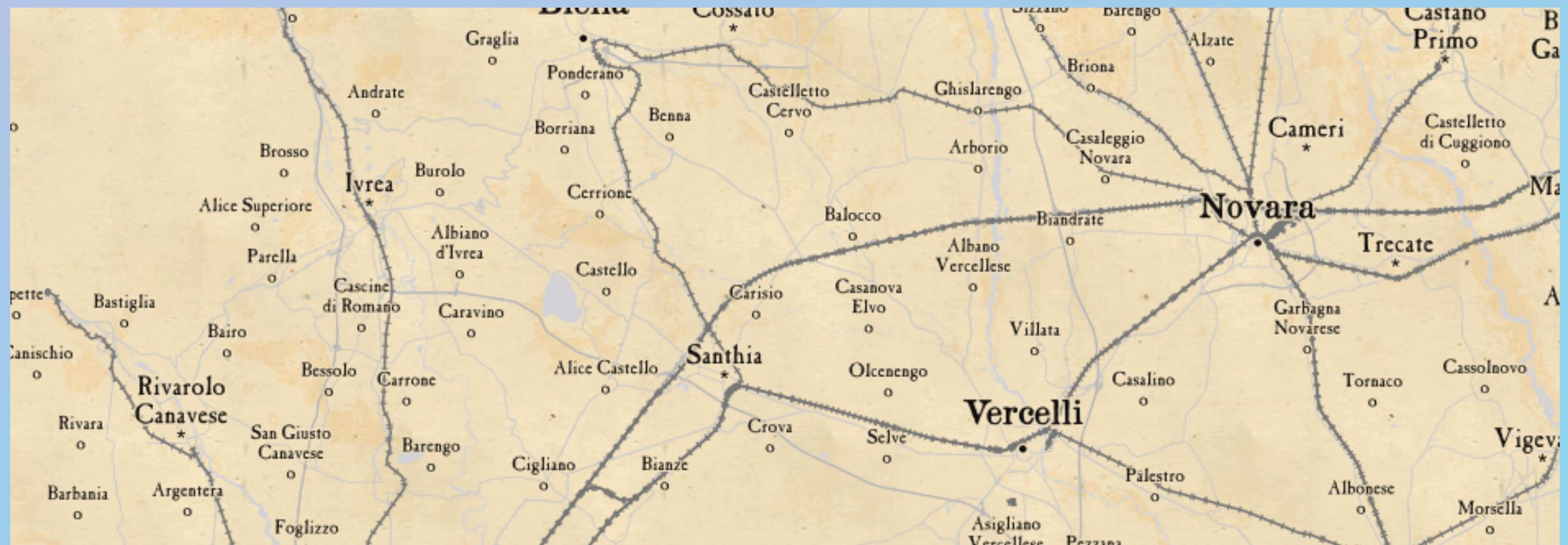


Basemaps
Points (Marker)
Lines/Polylines
Shapes (Polygons)
Images (Raster)
Controls

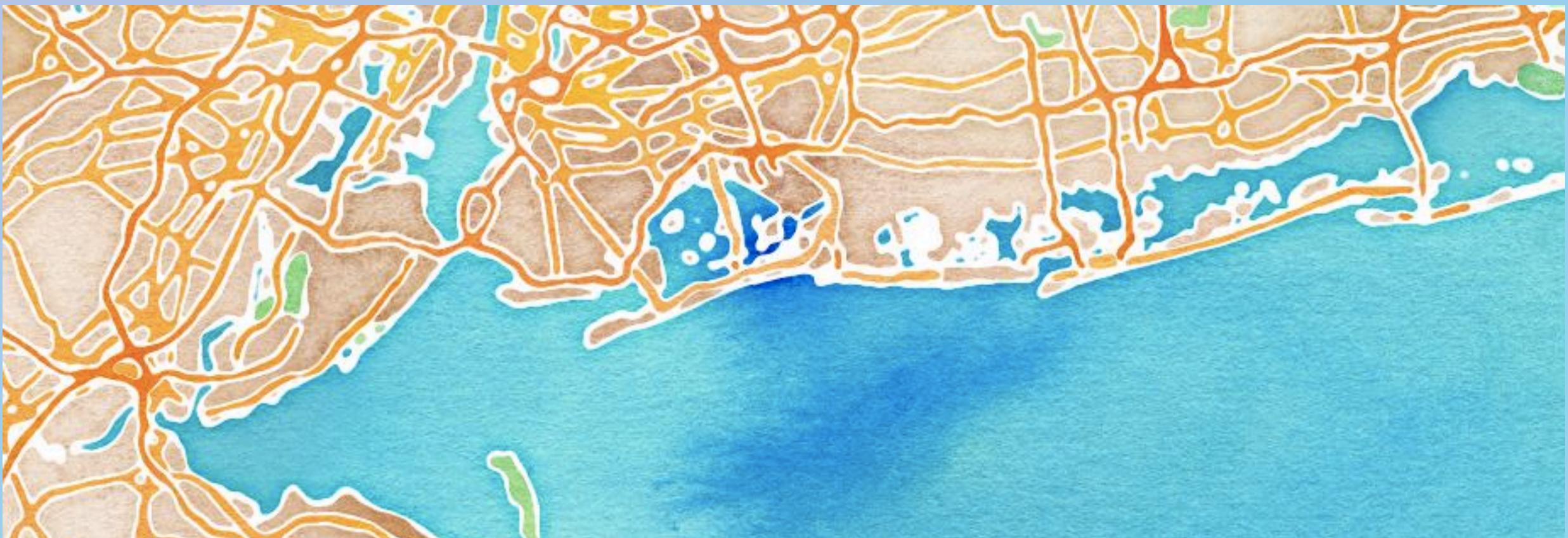
```
L.tileLayer('http://{s}.tile.openstreetmap.org/{z}/{x}/{y}.png', {  
    attribution: 'Map data © OpenStreetMap',  
    maxZoom: 18,  
    id: 'mapbox.streets',  
    accessToken: 'your.mapbox.access.token'  
}).addTo(myMap);
```



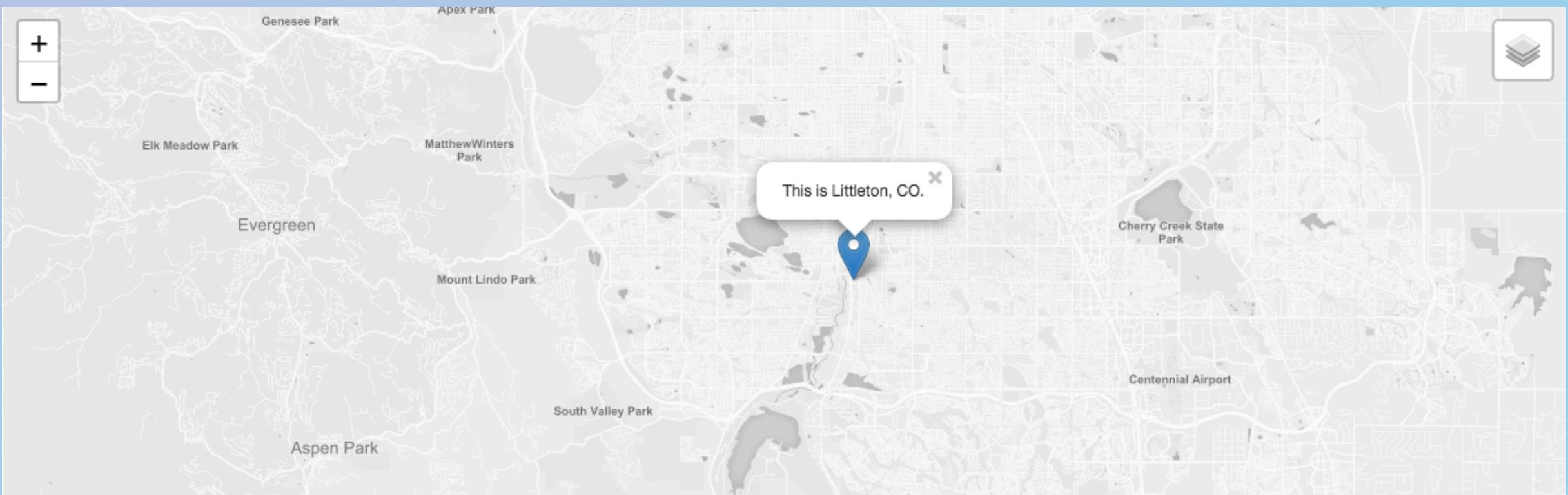
```
L.tileLayer('http://{s}.tile.thunderforest.com/pioneer/{z}/{x}/{y}.png', {  
    attribution: 'Map data &copy; OpenStreetMap',  
    maxZoom: 22,  
    apikey: 'XXXXXXXXZZZ',  
}).addTo(mymap);
```



```
L.tileLayer('http://stamen.ssl.fastly.net/watercolor/{z}/{x}/{y}.{ext}', {  
    attribution: 'Map data © OpenStreetMap',  
    maxZoom: 22,  
    apikey: 'XXXXXXXXXXXX',  
}).addTo(mymap);
```



```
// Manually add a marker with a popup  
littleton = L.marker([39.61, -105.02]).bindPopup('This is Littleton, CO.')  
littleton.addTo(map)
```



```
// Create a red polygon from an array of LatLng points
var latlngs = [[37, -109.05],[41, -109.03],[41, -102.05],[37, -102.04]];
var polygon = L.polygon(latlngs, {color: 'red'}).addTo(map)
```



BUT WHERE DOES THE DATA
COME FROM?

```
// Create a red polygon from an array of LatLng points
var latlngs = [[37, -109.05],[41, -109.03],[41, -102.05],[37, -102.04]];
var polygon = L.polygon(latlngs, {color: 'red'}).addTo(map)
```



JAVASCRIPT LOVES
DICTIONARIES OF OPTIONS

AND
CALLBACKS

```
function draw(geojson) {  
  // let's just put it on the map!  
  var dataLayer = L.geoJson(geojson)  
  dataLayer.addTo(map)  
}  
  
// pull in the geojson, then call 'draw'  
$.get('my-geojson-file.json', draw)
```

```
function draw(geojson) {
  // let's send some options to L.geoJson!
  // let's give each shape a color!
  // options at http://leafletjs.com/reference-1.1.0.html#path-option
  var dataLayer = L.geoJson(geojson, {
    style: function(feature, layer) {
      if(layer.properties.dem_votes > layer.properties.rep_votes) {
        return { fill: 'blue' }
      } else {
        return { fill: 'red' }
      }
    })
  dataLayer.addTo(map)
}

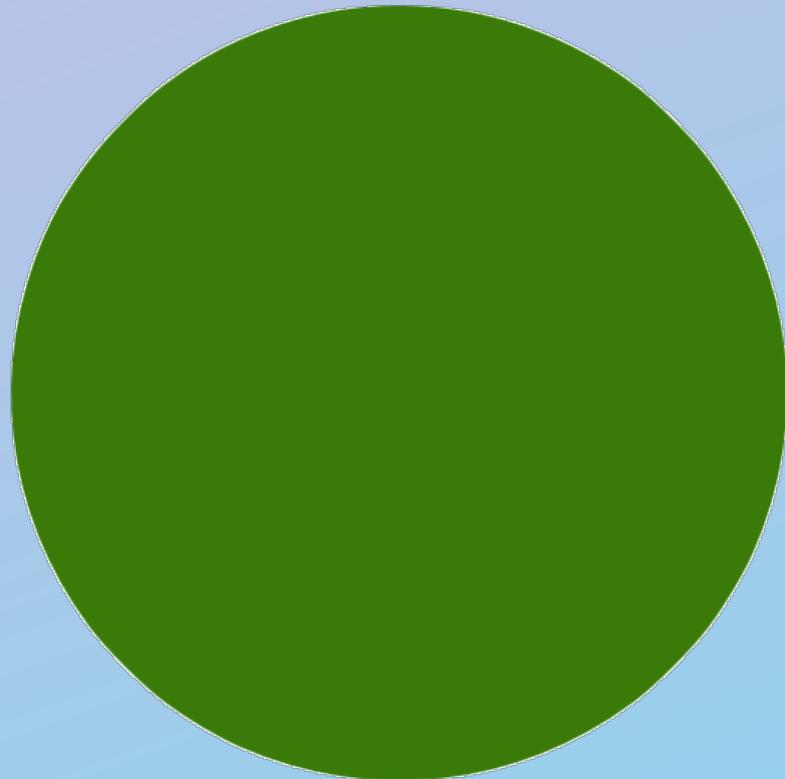
$.get('my-geojson-file.json', draw)
```

```
function draw(geojson) {  
  // let's send some options to L.geoJson!  
  // let's add a popup for when we click a shape!  
  // "go through every feature and do this thing"  
  
  var dataLayer = L.geoJson(geojson, {  
    onEachFeature: function(feature, layer) {  
      var popupText = "This is" + feature.properties.state_name;  
      layer.bindPopup(popupText);  
    }  
  })  
  dataLayer.addTo(map)  
}  
  
$.get('my-geojson-file.json', draw)
```

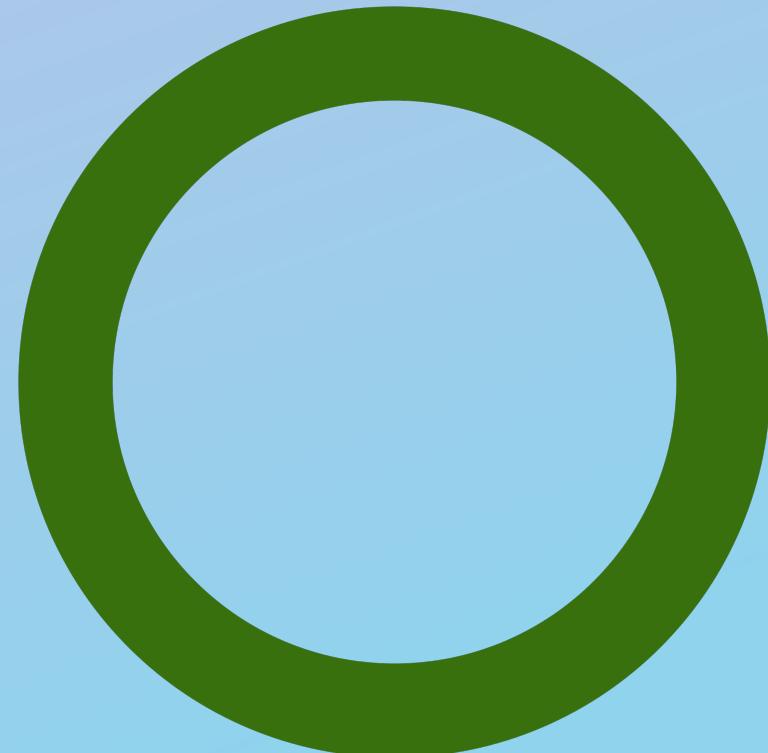
EXCELLENT DOCUMENTATION!

<http://leafletjs.com/reference-1.1.0.html>

fill



stroke



METHODS OF INTERACTION

Controls

- Zoom

On your map elements

- Click
- Hover

On the page

- Clicking to highlight/filter
- Dropdown to highlight/filtering

Zoom to fit/Center on something (fitBounds)

**JUST THINK ABOUT IT
PIECE BY PIECE!**

WHILE WE'RE AT IT: GOOD PLUGINS

Leaflet Omnivore will read in all sorts of formats (including CSV) and make everything easy to add to the map.

Leaflet Markercluster is for when you have a lot of points in a small area – it clusters them together until you zoom in!

OKAY OKAY OKAY
LET'S MAKE SOME STUFF